Climate In a Box User's Guide to Models

Robert Burns, Carlos Cruz, Shawn Freeman, Philip Hayes, Ramon Linan, Raj Pandian, Rahman Syed, Bruce VanAartsen and Gary Wojcik

Software Integration and Visualization Office
NASA Goddard Space Flight Center
Greenbelt, MD 20771

September 28, 2010

Contents

Chapter 1

System Overview

Thank you for your participation in the Climate in a Box (CIB) project! At NASA's Software Integration and Visualization Office (SIVO) we hope to create a system that will make climate and weather research far more accessible to the broader community. Although the project currently serves users with a basic software toolkit, we're moving forward with improving the user experience of data and code sharing.

With the Climate Toolkit, our focus has been more on software than hardware. Although we're moving toward our goal for any user to deploy the CIB Toolkit on any cluster with minimum hassle, for now we're delivering the software pre-installed onto new "desktop supercomputers" from SGI and Cray.

This User's Guide will provide CIB participants with a detailed set of instructions on how to administer a desktop supercomputer configured by SIVO, how to execute the models that have been installed, and how to configure new experiments moving forward.

1.1 Toolkit platform: hardware

The desktop supercomputers available so far have had a configuration similar to this, details may not be exact for all systems:

- 1 chassis with 8-10 "blades" or "nodes"

- Node configuration
  - 2 quad-core Intel Xeon cpus (8 cores total)
  - 24gb RAM
  - 300gb hard drive
- Internal InfiniBand network for high-speed MPI communication
- Internal Gigabit Ethernet network for administration
- External Gigabit Ethernet interface on headnode for Internet connectivity

1.2 Toolkit platform: software

The strategy for the Climate Toolkit is to emulate the major software components found on the NASA Center for Climate Simulation (NCCS) "Discover" supercomputing cluster. Developers benefit from having an environment similar across platforms, so any porting effort is made easier.

For the first version of the Climate Toolkit, SIVO has selected CentOS 5.5, a community-supported version of RedHat Enterprise Linux. With the OSCAR cluster toolkit and Torque scheduling system, this provides a solid platform for high performance computing (HPC) development. More information about these software packages can be found at these links:

http://www.centos.org/
http://svn.oscar.openclustergroup.org/trac/oscar

The Climate Toolkit will be based on versions 9.1, 10.1, and 11.1 of Intel's Compiler Suite (for C/C++/Fortran). To reduce complexity we have chosen to support only the mvapich2 MPI library, so the models and software become more manageable.

We have also installed the NASA Workflow Server, to make model setup and execution easy for new users. Users are encouraged to develop their own workflows to make management and preservation of different experiments much simpler.

For a full list of software and versions, please see Appendix A: Climate Toolkit Software Packages.

Chapter 2

NASA Modeling Guru Knowledge Base and Forums

NASAs Modeling Guru website is a user-friendly tool that allows those involved in Earth science modeling and research to find and share relevant information. The system hosts a growing repository for expertise in running Earth science models and using associated high-end computing (HEC) systems. It is open to the public, allowing collaboration with scientists, professors, and students from external research institutions.

This Web 2.0 site offers:

* Topic-based communities for various scientific models, software tools, and computing systems (i.e., Climate in a Box community)
* Each community has its own discussion forums, wiki-type documents, and possibly blogs
* E-mail notifications or RSS feeds for desired communities or blogs
* Homepage customization with drag-and-drop widgets
* Publicly available at: https://modelingguru.nasa.gov

If you register for an account, youll be able to post questions, receive email notifications, and customize you homepage. View registration instructions by clicking Request an Account at the top of the page.*NCCS account-holders

Figure 2.1: Climate in a Box Community in the NASA Modeling Guru Web Site.

(for Discover, Dirac, etc.) are preregistered – just login with your current username and password.

For CIB users, most relevant resources would be the found in the Climate in a Box community, which can be found by clicking its link under the Communities menu on the homepage, or by using this direct URL: https://modelingguru.nasa.gov/community/systems/climateinabox

Figure 2.1 shows a portion of the Overview page for the Climate in a Box Community.

Note that the tabs at the top let you select a specific content type to view. Documents include information like model run procedures or benchmark results, while Discussions are your forum for posting questions. If youd like to be notified about all new posts to this community, click Receive email notifications on the right (must be logged in to see this option).

If you cant find information you need on the site, click Start a Discussion to post your question. Moderators will monitor this area and will normally respond the same day. We also expect to have a live-chat support feature, which should be available most days during business hours.

Chapter 3

Torque Scheduler

The Torque Scheduler is software that allows for several users on a cluster to share the nodes efficiently. It is similar in operation to the PBS Pro software used on the NCCS Discover cluster. Users recognize it as the "qsub" command.

For extensive documentation, please visit the Torque website:

`http://www.clusterresources.com/products/torque-resource-manager.php`

Most end users will find all they need here:

`http://www.clusterresources.com/products/torque/docs/2.1jobsubmission.shtml`

Users can create scripts that launch MPI tasks and submit them in the background for execution. Queues are available so that users can reserve the system for their experiment and release it others when their job finishes. The following directives should be placed at the top of a user script:

```
#PBS -S /bin/bash
#PBS -N somejobname
#PBS -l nodes=2:ppn=8
#PBS -l walltime=00:30:00
```

Once the user script has been created that loads the right application modules, cd to the correct directory, then a user can submit this job to the queue by typing "qsub run_script.bash"

Submitting a script to the background is one way of using the computational resources on these clusters. Another useful way is called an "interactive PBS session" which can help with debugging problems. This method allows a user to have their terminal "land" on a compute node where he can launch MPI applications directly. Here's a sample session:

```
[someuser@headnode run]$ hostname
headnode.cx1
[someuser@headnode run]$ qsub -I -l nodes=2:ppn=8,walltime=00:30:00
qsub: waiting for job 1505.headnode.cx1 to start
qsub: job 1505.headnode.cx1 ready
[someuser@node-7 ~] hostname
node-7.cx1
[someuse@node-7 ~]$ mpirun_rsh -np 16 ./somemodel.x
```

With an interactive session, a script is not provided to the qsub command. Instead, a user must use the "-I" flag to indicate that it is an interactive session, as well as the PBS parameters to indicate how many resources he would like to use.

Users must remember to type "exit" to leave an interactive session when they're finished; otherwise resources might sit idly while other users cannot access them.

Chapter 4

NASA Workflow Tool

4.1 Tools and Data

The workflow tool is set up on the CIB system under the following paths:

* **/cib/workflow_tool**-Containstheinitialworkflowconfigurations,workflow datasets, experiment runs, and other workflow-specific information.
* **/cib/outputdata/<username>/workflow** -Typicallycontainsthelargeoutputs generated by workflows. Note that Generic User mode in NED places outputs under the /cib/outputdata/workflow/workflow folder.
* **/optand/cib/libraries** - Contains various tools and applications required to run the workflow tool or individiual workflows.
* **/opt/NEDand/cib/libraries/workflow** - Contains the **NASAExperiment Designer** client, server and workflow editing software.

4.2 General Operations

The workflow tool requires three processes to be launched during system reset. These processes should generally be run by the dedicated workflow account.

To start the NED server:

```
nohup /opt/NED/server_deployment/runNedServer.sh > ~/
nedserverNohup.log
```

To start the eXist database, which is used to track experiments:

```
nohup /opt/eXist/bin/startup.sh > ~/existNohup.log
```

To create a tunnel to the GEOS5 revision control server:

NOTE: Currently, the tunnel to cvsacl is restricted and requires dual-factor authentication. Because of this, a tunnel must be set up, using an NCCS passcode. Due to NCCS policy, this tunnel will currently only stay active for 7 days. This is only required for checkouts of GEOS5 with the workflow.)

```
nohup ssh username@cvsacl > nohup_cvsacl.log
```

After this, you will be prompted for your passcode and your password. Once you have entered the password, you can press CTRL Z to background the tunnel. To ensure that the tunnel has connected properly, you can check the log file output.

4.3 User Setup

At a bare minimum, all users that will be running workflows must belong to the user group that will correspond to running workflows. Currently, this group is the CIB group. Admins of the box may add additional groups for controlling access to different workflow directories.

Assuming that a global profile has been established following the installation instructions, no additional user setup is required for running general workflows.

If a user or users wish to run workflows as themselves instead of using the generic workflow account, then the user would be required to add the public SSH key of the workflow account running the server into the users SSH authorized_keys file.

4.4 Workflow Tool Installation

Please refer to the NASA Workflow Tool: Administrator Guide section on deploying the NED server for information about setting up and installing the workflow tool and its dependencies.

Chapter 5

GEOS-5 AGCM

The Goddard Earth Observing System (version 5) suite and its Atmospheric General Circulation Model (AGCM) are being developed at the Global Modeling and Assimilation Office (GMAO) to support NASA's earth science research in data analysis, observing system modeling and design, climate and weather prediction, and basic research.

The AGCM is a weather-and-climate capable model being used for atmospheric analyses, weather forecasts, uncoupled and coupled climate simulations and predictions, and for coupled chemistry-climate simulations. There are model configurations with resolutions ranging from 2^o to $1/4^o$, and with 72 layers up to 0.01 hPa, resolving both the troposphere and the stratosphere.

The AGCM structure consists of a hierarchy of ESMF gridded components ([1]) constructed and connected using MAPL. MAPL is a software layer that establishes usage standards and software tools for building ESMF compliant components. The model's architecture is illustrated in figure 5.1. Most of the calculations are done by components at the leafs of this tree. The composite components above the leafs serve as connectors, performing specialized connections between the children and producing combined outputs.

Parallelization is primarily implemented through MPI, although some key parts of the code, such as the model dynamics, also have Open-MP capability. The model runs on a 2-D decomposition, transposing internally between horizontal and vertical layouts. Some of the physics such as the solar radiation, which at any given time is active over only half the globe,

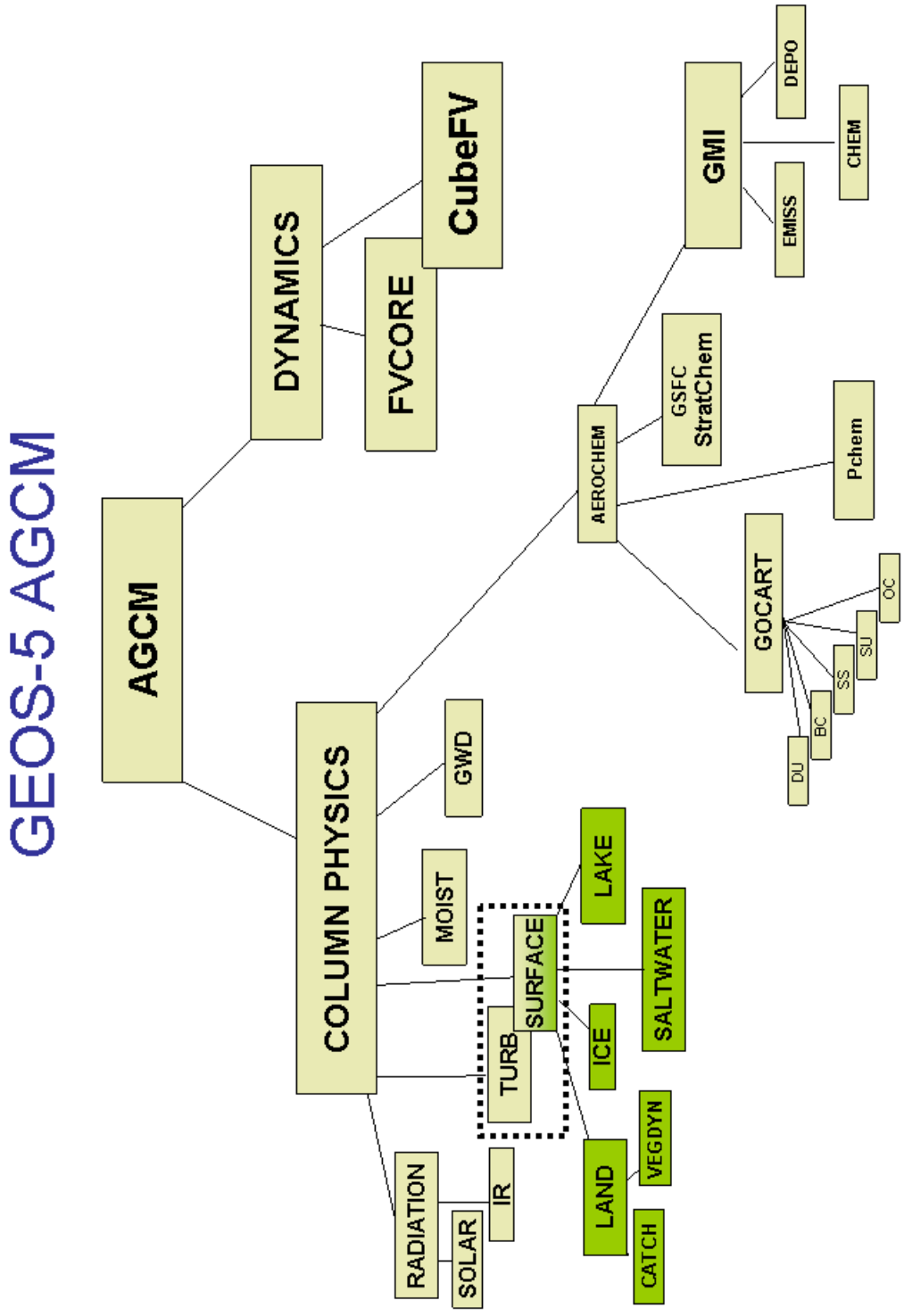

Figure 5.1: The GEOS-5 AGCM structure.

is load balanced. The code scales well across compute nodes and scalability increases linearly with problem size.

For additional information see the GMAO GEOS5 web site.

5.1 Configuring and building

The GEOS-5 AGCM can be potentially built on a variety of platforms, including SGI, IBM, Darwin and Linux systems. However most of our testing and portability tools are limited to a handful of Linux systems. In the following sections we provide information on setting up and running one version of the AGCM: Fortuna 2.1. Other versions of GEOS-5 may differ in details that, while minor, will require different steps for building and running, so *caveat utilitor*.

The following description assumes that you know your way around Unix, have successfully logged into your NCCS account (presumably on the DISCOVER cluster) and have an account on progress with the proper ssh preparation (see the progress repository quick start). However the same commands can be used in the CIB system unless otherwise noted.

The commands below assume that your shell is csh. This is for convenience since the scripts to build and run GEOS-5 tend to be written in the same shell. If you prefer or use a different shell, it is easiest just to open a csh process to build the model and your experiment.

5.1.1 MERRA vs Fortuna Tags

MERRA refers to the NASA reanalysis data set that uses a major new version of the GEOS-5 Data Assimilation System ([3]). In the context of Climate In a Box (CIB), MERRA is that and, additionally, it refers to the particular AGCM code base (tag) used in the generation of the MERRA data set[1].

The newest set of code releases are named Fortuna and the latest stable tag is Fortuna 2.1 (henceforth referred to as simply Fortuna).

The CIB toolkit is loaded with both the MERRA and Fortuna tags. The next section describes the minimum steps required to build and run GEOS-

[1]The tag, GEOSagcm-Eros_7_25, was part of the Eros family of code releases

5 Fortuna 2.1 on a Linux platform. You should successfully complete the steps in these instructions before doing anything more complicated. A CIB quick start guide that uses the MERRA/FORTUNA stand-alone packages is described in section (section **??**).

5.1.2 Baselibs

Baselibs are a set of support libraries (ESMF, HDF, netCDF, etc) that need to be installed before attempting to build the GEOS-5 AGCM application. Note that the CIB toolkit will contain preinstalled Baselibs for both the MERRA and Fortuna tags. In this section we provide a brief description of Baselibs installation.

Note that the following compilers and utilities are required for compiling and building the Baselibs

- Fortran90 (or later) compiler;
- C compiler
- MVAPICH2, openMPI or intel MPI compatible with the above compilers
- GNU make
- Perl - to run some scripts
- LATEX - to produce documentation

and the same should be used to build the Fortuna tag.

Building the Baselibs from scratch requires an account on NASA-GSFC NCCS systems to access the progress repository as well as permission to read the baselibs repository.

When you have permission to checkout the Baselibs software from the progress repository, login to your target system and issue these commands:

```
setenv CVS_RSH ssh
cvs -d :ext:USERNAME@progress.nccs.nasa.gov:/cvsroot/baselibs
        checkout -r SOMETAG Baselibs
```

Please replace USERNAME with your LDAP userid. Please replace SOMETAG with a valid Baselibs tag. If you are using GEOS, consider tag "GMAO-Baselibs-3_2_0", which is compatible with the Fortuna 2.1 tag. Also please note that the module name is "Baselibs", case sensitive.

Other available tags:

```
GMAO-Baselibs-3_1_4 (use with Fortuna 1.x tags).
GMAO-Baselibs-2_1_1 (use with MERRA tag).
```

For each system, the environment must be setup so the build process will use the desired compilers. The Baselibs are generally portable but tweaks may be needed and their discussion is beyond the scope of this manual. Therefore it is recommended that the user refer to NASA Modeling Guru web site.

When issuing the make commands found in the INSTALL document, it's best to route all output into a log file for debugging purposes. Use the following conventions to capture all information and run the installation in the background:

To install:

```
make install >& install.log &
```

You can also issue a "tail -f install.log" to watch the process run to completion if you'd like.

When the process has completed, search install.log for the keyword "Error". Ignore any "Error" string that's found in a filename or as part of a compiler warning. If there are legitimate errors, please review your environment setup and build command prior to contacting support.

After installation is complete the environment variable BASEDIR specifying the path of the Baselibs must be set.

5.2 Installing the Fortuna Tag

Set the following environment variable:

```
setenv CVSROOT :ext:USERID@progress.nccs.nasa.gov:/cvsroot/esma
```

where USERID is, of course, your progress username, which should be the same as your NASA and NCCS username. Then, issue the command:

```
cvs co -r  Fortuna-2_1 Fortuna
```

This should check out the Fortuna version of the model from progress and create a directory called GEOSagcm. If you cd into GEOSagcm/src there is a script, g5_modules, that can be used to setup the proper computational environment on NCCS systems; otherwise it will likely not be portable. If you are working on NCCS systems

```
source g5_modules
```

Else, on the CIB system, use the script provided under /cib/libraries/archives

```
cp /cib/libraries/archives/set_mode.<your shell> ~/
```

To use execute the script and it will provide usage instructions. When done type:

```
module list
```

and you should see:

```
Currently Loaded Modulefiles:
1) comp/intel-11.1
2) mpi/mvapich2-1.5/intel-11.1
```

And the following environment variables must be set

```
setenv BASEDIR <PATHS WHERE BASELIBS ARE INSTALLED>
setenv LD_LIBRARY_PATH $LIBRARY_PATH:$BASEDIR/$arch/lib
```

The environment variable BASEDIR was described in the previous section. The environment variable LD_LIBRARY_PATH is a colon-separated set of directories where libraries should be searched for first, before the standard set of directories. On the MAC this variable is called DYLD_LIBRARY_PATH. On the CIB system the set_mode script will take care of this.

If this all worked, then type:

```
gmake install > & makefile.log &
```

This will build the model and redirect the gmake output to the makefile.log file. You can check the gmake progress by typing "tail -f makefile.log". The installation will take about 45 minutes. If this works, it should create a directory under GEOSagcm called Linux/bin. In here you should find the executable: GEOSgcm.x.

5.3 Input data/Boundary conditions

NOTE: To obtain data from outside the NCCS firewall users must log in to the data migration facility (DMF). The NCCS is currently running DMF on an SGI Origin 3000 called palm, and the host name for the public-facing system is dirac.gsfc.nasa.gov.

What input data (ICs, BCs, resource files) do we need to run a GEOS-5 AGCM experiment? Generally, model input - including restarts and resource files - is associated with a particular CVS "tag", like MERRA and Fortuna. The CIB toolkit provides data sets to be used with the MERRA and Fortuna tags at 2^o resolution. Currently there is no easy way to gather up all the necessary data to run a particular experiment at a specific resolution. If you need a specific data set please contact SIVO.

5.4 Setting Up An Experiment

First of all, to run jobs on the cluster you will need to set up passwordless ssh (which operates within the cluster). On the CIB system there is nothing to do. On the NCCS cluster run the following from your home directory:

```
cd .ssh
ssh-keygen -t dsa
cat id_dsa.pub >> authorized_keys
```

Similarly, if working on the NCCS cluster, transferring the daily output files (in monthly tarballs) requires passwordless authentication from DISCOVER to PALM. While in ∼/.ssh on DISCOVER, run

```
ssh-keygen -t dsa
```

Then, log into PALM and cut and paste the contents of the id_rsa.pub and id_dsa.pub files on DISCOVER into the ∼/.ssh/authorized_keys file on PALM. Problems with ssh should be referred to NCCS support.

To set the model up to run we need to create an experiment directory (scratch) that will contain GEOSgcm.x, restarts, BCs and resource files. On the CIB system the system can be setup using the experiment setup tool contained in /cib/models/archive. These are the steps to follow:

```
cp /cib/models/archive/AGCMsetup.tgz /cib/outputdata/<userid>
tar xfz AGCMsetup.tgz
cd AGCMsetup
EDIT options.rc file
RUN setupg5gcm
```

AGCMsetup helps setup MERRA or Fortuna run (2^o resolution only) but we still need to create a run script. A sample one, fortuna_run.j, can be obtained from /cib/models/archives with only a minor edit necessary.

On NCCS systems the GEOSagcm/src/Applications/GEOSgcm_App directory contains the script gcm_setup than can also be adapted to the CIB system. To run:

```
gcm_setup
```

The gcm_setup script asks you a few questions such as an experiment name (with no spaces, called EXPID) and description (spaces ok). It will also ask you for the model resolution, expecting one of the available lat-lon domain sizes, the dimensions separated by a space. Here are some details regarding the required input:

- The Experiment ID should be an easily identifiable name representing each individual experiment you create. One possibility is your initials followed by a 3-digit number: abc001
- The Experiment Description should be a concise statement describing the relevant nature of your experiment.

- Model Resolution Options:

```
 72  46   (~ 4 -deg)
144  91   (~ 2 -deg)
288 181   (~ 1 -deg)
576 361   (~1/2-deg)
1152 721  (~1/4-deg)
```

- The AERO PROVIDER describes the Chemistry Component to be used for Radiatively Active Aerosols. PCHEM is the default used for most AMIP-style Climate Runs.

- The HOME directory will contain your run scripts and primary resource (rc) files.

- The EXP directory will contain your restarts and model output. The HOME and EXP directories can be the same, e.g. /discover/nobackup/$LOGNAME/abc001

- Your GROUP ID is your charge code used for NCCS accounting (not required on CIB).

For your first time out you will probably want to enter 144 91 (corresponding to ∼2 degree resolution). Towards the end it will ask you for a group ID. Enter whatever is appropriate, as necessary. The rest of the questions provide defaults which will be suitable for now, so just press enter for these. Note that gcm_setup will not work outside of NCCS systems, however the gcm_setup included with the standalone package (section **??**) will work on the CIB platforms.

The script produces an experiment directory (EXPDIR) in your space (on DISCOVER) as /discover/nobackup/USERID/EXPID, which contains, among other things, the sub-directories:

- post (containing the post-processing job script)

- archive (containing an incomplete archiving job script)

- plot (containing an incomplete plotting job script)

The post-processing script will complete (i.e., add necessary commands to) the archiving and plotting scripts as it runs. The setup script that you ran also creates an experiment home directory (HOMEDIR) as ∼USERID/geos5/EXPID containing the run scripts and GEOS resource (.rc) files.

The run scripts need some more environment variables, here are the minimum contents of a .cshrc:

```
umask 022
unlimit
limit stacksize unlimited
set arch = ‘uname‘
```

The umask 022 is not strictly necessary, but it will make the various files readable to others, which will facilitate data sharing and user support. Your home directory ∼USERID is also inaccessible to others by default; running chmod 755 ∼ is helpful.

Copy the restart (initial condition) files and associated cap_restart into EXPDIR. Keep the "originals" handy since if the job stumbles early in the run it might stop after having renamed them. The model expects restart filenames to end in "rst" but produces them with the date and time appended, so you may have to rename them. The cap_restart file is simply one line of text with the date of the restart files in the format YYYYMMDD<space>HHMMSS. The boundary conditions/forcings are provided by symbolic links created by the run script. If you need an arbitrary set of restarts please contact SIVO.

The script you submit, gcm_run.j, is in HOMEDIR. It should be ready to go as is. The parameter END_DATE in CAP.rc (previously in gcm_run.j) can be set to the date you want the run to stop. You may edit the .rc files directly instead of template (.tmpl). An alternative way to stop the run is by commenting out the line if ($capdate < $enddate) qsub $HOMEDIR/gcm_run.j at the end of the script, which will prevent the script from being resubmitted, or rename the script file. You may eventually want to tune parameters in the CAP.rc file JOB_SGMT (the number of days per segment, the interval between saving restarts) and NUM_SGMT (the number of segments attempted in a job) to maximize your run time.

Submit the job with qsub gcm_run.j. On DISCOVER you can keep track of it with qstat or qstat — grep USERID, or stdout with tail -f /DISCOVER/pbs_spool/JOBID.OU, JOBID being returned by qsub and dis-

played with qstat. Jobs can be killed with qdel JOBID. The standard out and standard error will be delivered as files to the working directory at the time you submitted the job.

5.4.1 RC Files

You need to edit a couple of files in HOMEDIR to get the experiment to run the way you want:

AGCM.rc

This file controls the model run characteristics. For example, a particular AGCM.rc file has the following lines characteristic (to run GOCART with climatological aerosol forcing):

```
GOCART_INTERNAL_RESTART_FILE:           gocart_internal_rst
GOCART_INTERNAL_RESTART_TYPE:           binary
GOCART_INTERNAL_CHECKPOINT_FILE:        gocart_internal_checkpoint
GOCART_INTERNAL_CHECKPOINT_TYPE:        binary
AEROCLIM:    ExtData/AeroCom/L72/aero_clm/gfedv2.aero.eta.%y4%m2clm.nc
AEROCLIMDEL: ExtData/AeroCom/L72/aero_clm/gfedv2.del_aero.eta.%y4%m2clm.nc
AEROCLIMYEAR: 2002
DIURNAL_BIOMASS_BURNING: no
RATS_PROVIDER: PCHEM   # options: PCHEM, GMICHEM, STRATCHEM
                       # (Radiatively active tracers)
AERO_PROVIDER: PCHEM   # options: PCHEM, GOCART
                       # (Radiatively active aerosols)
```

To run with interactive aerosol forcing, modify the appropriate lines above to look like:

```
#AEROCLIM:    ExtData/AeroCom/L72/aero_clm/gfedv2.aero.eta.%y4%m2clm.nc
#AEROCLIMDEL: ExtData/AeroCom/L72/aero_clm/gfedv2.del_aero.eta.%y4%m2clm.nc
#AEROCLIMYEAR: 2002
AERO_PROVIDER: GOCART   # options: PCHEM, GOCART
                        # (Radiatively active aerosols)
```

HISTORY.rc

This file controls the output streams from the model. These are know as collections:

```
COLLECTIONS: 'geosgcm_prog'
             'geosgcm_surf'
             'geosgcm_moist'
             'geosgcm_turb'
             'geosgcm_gwd'
             'geosgcm_tend'
             'geosgcm_bud'
             'tavg2d_aer_x'
             'inst3d_aer_v'
             ::
```

Additional information about the HISTORY.rc file can be found in this link.

CAP.rc

Among other things, this file controls the timing of the model run. For example, you specify the END_DATE, the number of days to run per segment (JOB_SGMT) and the number of segments (NUM_SGMT). If you modify nothing else, the model would run until it reached END_DATE or had run for NUM_SGMT segments.

cap_restart

This file doesn't exist here by default, so create it with your favorite editor. As explained earlier, cap_restart will have the model start date (ignore BEG_DATE in CAP.rc; if this file exists it supersedes that). It should just specify YYYYMMDD HHMMSS start time of model run. For example:

```
19991231 210000
```

Additional information can be found here and at the GEOS5 wiki site.

5.5 Output and Plots

During a normal run, the gcm_run.j script will run the model for the segment length (current default is 8 days). The model creates output files (with an nc4 extension), also called collections (of output variables), in EXPDIR/scratch directory. After each segment, the script moves the output to the EXPDIR/holding and spawns a post-processing batch job which partitions and moves the output files within the holding directory to their own distinct collection directory, which is again partitioned into the appropriate year and month. The post processing script then checks to see if a full month of data is present. If not, the post-processing job ends. If there is a full month, the script will then run the time-averaging executable to produce a monthly mean file in EXPDIR/geos_gcm_*. The post-processing script then spawns a new batch job which will archive (on DISCOVER) the data onto the mass-storage drives (/archive/u/USERID/GEOS5.0/EXPID).

If a monthly average file was made, the post-processing script will also check to see if it should spawn a plot job. Currently, our criteria for plotting are:

```
1) if the month created was February or August, AND
2) there are at least 3 monthly average files,
   then a plotting job for the seasons DJF or JJA will be issued.
```

The plots are created as gifs in EXPDIR/plots. The post-processing script can be found in:

```
GEOSagcm/src/GMAO_Shared/GEOS_Util/post/gcmpost.script.
```

The nc4 output files can be opened and plotted with gradsnc4 but use sdfopen instead of open. See the Grads web site for a tutorial.

The contents of the output files (including which variables get saved) may be configured in the HOMEDIR/HISTORY/tmpl – a good description of this file may be found on Modeling Guru.

5.6 Case Study

The setup provided in AGCMsetup.tgz is intended to produce an experimental setting with the following characteristics:

- Resolution 144x91x72 levels
- 5-day simulation with starting date/time 19920110 210000
- Comprehensive prognostic and diagnostic output over the simulation period.
- 16cpu run (2 nodes)

The case study can be used as a benchmarking suite to port the model among platforms and to examine model reproducibility (see for example this document) and to some extent performance (see this document).

Chapter 6

NASA-GISS ModelE

The most current version of the NASA-GISS series of coupled atmosphere-ocean models is called ModelE. The ModelE source code can be down- loaded from the GISS web site. Model documentation, including the ModelE specification and results from three standard configurations, is given in [4].

The CIB system comes with a pre-installed version of modelE (8/10/2010 snapshot). In this chapter we describe the steps required to build, install and run modelE on the CIB system. The CIB system also contains a stand-alone package to help expedite the experiment setup and is described in section 6.5.

6.0.1 System requirements

System requirements are the following:

- Fortran 90 (or later) compiler;
- C compiler
- MVAPICH2, openMPI or intel MPI compatible with the above compilers
- GNU make (version 3.79 or higher)
- Perl (version 5.005 or higher)

ModelE has been tested with the SGI, IBM, and COMPAQ/DEC compilers on their respective workstations. For Linux, Macs or PCs, the choice of compiler is wider.

Parallelization is primarily implemented through MPI but some parts of the code also have Open-MP capability.

Many GCM subroutines declare temporary arrays which, in FORTRAN, reside in a part of the computer's memory known in computer parlance as the "stack." It is essential that a user's stack size be large enough to accommodate these arrays; a minimum value of 8192 kilobytes is recommended, with stack requirements increasing with model resolution. Default stack sizes vary among different computing environments, but can be easily adjusted. Unfortunately this parameter must be explicitly set by the user rather than by the GCM setup scripts. Stack size is set by commands native to a user's $SHELL; for example, a bash user would type "ulimit -s 32768" to get a stack size of 32768 kb. Mysterious crashes in the radiation are quite often related to a too small stack.

6.1 Configuration and Building

6.1.1 Getting the code

To build/run the latest modelE code base you will need to check it out from the GISS repository. That requires an account on the GISS server. Please contact contact Gavin Schmidt at GISS.

After you have gained access to the server you can checkout the code as follows:

```
export CVSROOT=simplex.giss.nasa.gov:/giss/cvsroot
export CVS_RSH=ssh
cvs co modelE
```

This will get you the latest code snapshot. The distribution of modelE has a top directory, named modelE, which contains the following sub-directories:

```
aux config decks E2000 init_cond model README tests
cmor CVS doc exec Makefile prtdag templates
```

The CVS directory is for version control purposes and may also be found in the sub-directories.

6.1.2 configuring modelE

The very first step in the modelE setup is to create of a file called .modelErc. This file keeps some system-wide information (such as your directory structure) which is common to all modelE runs you do on current machine. To create this file go to the directory modelE/decks and execute:

```
gmake config
```

Note file name requires a "." dot. This file must be placed on your home directory. The file contains global options for modelE some of which must be modified. The following is a sample .modelErc file:

```
# This file contains global options for modelE.

## Directory structure ##

# DECKS_REPOSITORY - a directory for permanent storage of run info.
# All rundecks that you create will be copied to this directory.
DECKS_REPOSITORY=/cib/outputdata/ccruz2/modelE.scratch/decks

# CMRUNDIR - directory to which all run directories will be linked.
# This directory will be searched by most scripts for locations of
# specific runs.
CMRUNDIR=/cib/outputdata/ccruz2/modelE.scratch/cmrun

# EXECDIR - path to directory with modelE scripts and with some
# executables. This directory should contain the scripts from modelE/exec.
EXECDIR=/cib/outputdata/ccruz2/modelE.scratch/exec

# SAVEDISK - a directory where all run directories (which will contain
# all output files such as rsf, acc etc.) will be created. This should
# be big enough to accomodate all model output.
SAVEDISK=/cib/outputdata/ccruz2/modelE.scratch/out
```

```
# GCMSEARCHPATH - directory to search for gcm input files.
# All necessary input files should be copied or linked to this directory.
GCMSEARCHPATH=/cib/inputdata/modele/input

# NETCDFHOME - path to location of netcdf library. Leave blank if netcdf
# is not installed on this computer
NETCDFHOME=/cib/libraries/netcdf

## Customizable setings for Makefile ##

# OUTPUT_TO_FILES - if set to YES all errors and warnings will be sent
# to files with the names <source_name>.ERR
OUTPUT_TO_FILES=YES

# VERBOSE_OUTPUT - if set to YES gmake will show compilation commands
# and some other information. Otherwise most of the output will be
# suppressed
VERBOSE_OUTPUT=YES

# Compiler. For options look under config/ directory for files named
# compiler.<COMPILER>.mk
COMPILER=intel

# MPI. # Compiler. For options look under config/ directory for files named
# mpi.<MPIDISTR>.mk
MPIDISTR=mvapich2
# This needs to be specified if installation is in non-standard directory
MPIDIR=/opt/mpi/mvapich2/1.5/intel-11.1

# ESMF
ESMF=NO
ESMF_COMM=mpich2
ESMF_BOPT=O
# path to location of ESMF installation.
ESMFINCLUDEDIR=/cib/libraries/esmf/include
ESMFLIBDIR=/cib/libraries/esmf/lib
```

```
# multiprocessing support. If set to YES gmake will compile the code
# with OpenMP instructions.
MP=NO
```

This file contains several entries that need to be described in more detail. Generally the directories specified in DECKS_REPOSITORY, CMRUNDIR, EXECDIR and SAVEDISK (decks, cmrun, exec and out respectively) have to be manually created. Therefore in the above example:

```
cd /cib/outputdata/ccruz2/modelE.scratch
mkdir decks cmrun exec out
```

GCMSEARCHPATH contains the input files needed for a particular experiment. The CIB system contains data for E1M20 and E4M20. More data can be obtained using the utility described in section 6.5. COMPILER and MPIDISTR are entries that must correspond to the corresponding files under modelE/config. For example if COMPILER=intel then there must be a file named modelE/config/compiler.intel.mk. Likewise if MPIDIST=mvapich2 then modelE/config/mpi.mvapich2.mk must exist. This is important or the model compilation will fail.

Note that there are entries for netCDF (NETCDFHOME) and ESMF. These are prerequisites in order for modelE to produce netCDF output and to run in parallel respectively. Therefore these entries must point to existing directories for a proper modelE installation. On the CIB system there are pre-installed versions of netCDF and ESMF under

```
/cib/libraries/netcdf/3.6.2_intel11
/cib/libraries/esmf/2.2.2rp3_intel
```

Both libraries were compiled with Intel11 making it necessary to use the same compiler when building modelE.

6.1.3 Model installation

To compile/build the model first setup your module environment, that is choose a compiler and an MPI implementation. For example on the CIB platform:

```
module load comp/intel-11.1 mpi/mvapich2-1.5/intel-11.1
```

In order to run the GCM a 'rundeck' file (with a suffix .R) must be created. This contains information about the run you would like to perform, the fortran code that you intend to use, pre-processing options that are required, the boundary and initial conditions and any run-specific parameters (such as the time step, length of run etc.). Most of the specifics have been automated using 'gmake'. The sub-directory 'decks' is the main control directory, most commands will be issued in this directory. Typing 'gmake' will produce some documentation. There are a number of sample rundecks in the templates/ sub-directory. Here we will use the E1M20 rundeck as a 'source' and create a copy named E1M20copy:

```
gmake rundeck RUNSRC=E1M20 RUN=E1M20copy
```

The list of fortran files within the rundeck determines which files are compiled into the executable. The compilation is automatic, and will take account of all dependencies, so that if a file is modified it will be re-compiled, along with any files that directly or indirectly depend on modules or routines within it. The executable is made using

```
gmake gcm RUN=E1M20copy
```

Note that this is a serial build by virtue of our choices in the .modelErc file, namely ESMF=NO. If this runs successfully there will be a message saying so.

Parallel Build

For a parallel build you will need to build ESMF prior to building modelE. ESMF refers to the Earth System Modeling Framework. For more information see this web site.

Note that modelE still uses a frozen version of ESMF, namely ESMF_2_2_2rp3, which can be downloaded from the ESMF web site. Additional information for installing ESMF can also be found there.

After installing ESMF edit your .modelE rc file and set ESMF=YES. Also change ESMFINCLUDEDIR and ESMFLIBDIR accordingly. Then rebuild modelE.

As mentioned earlier, ESMF is installed under

```
/cib/libraries/esmf/2.2.2rp3_intel
```

6.2 Input data/Boundary conditions

GISS has set up a repository for modelE input files on NCCS machine called "dataportal". This files are available for public download from any computer by accessing the related dataportal modelE page:

```
http://portal.nccs.nasa.gov/GISS_modelE/
```

One can also use updated script modelE/exec/get_input_data for automatically downloading all input files necessary to work with a particular rundeck. The typical command to do this:

```
get_input_data -w <my_rundeck>  <data_dir>
```

where <my_rundeck> is the name of the rundeck you are working with and <data_dir> is the directory where you want to put the downloaded input files. To update your local repository you may want to point <data_dir> to the directory $GCMSEARCHPATH specified in your /.modeErc (get_input_data will skip existing files).

Currently the repository on Data Portal contains all input files necessary to work with all rundecks in modelE/templates.

6.3 Executing an experiment

To actually run the model, you must first 'setup' the run by running the first hour (or specifically the first source time step).

```
gmake setup RUN= E1M20copy
```

Alternatively you may want to use

```
gmake setup_nocomp RUN= E1M20copy
```

"setup_nocomp" means that the code will not be recompiled. This is not the case if one uses "gmake setup". If this runs successfully there will be a message saying so.

As is, the "gmake setup_nocomp" command runs the model for 1 hour to simply set up a model run but interesting experiments must run for more than 1 hour. One way to do it is by editing the rundeck file. For the earlier example the file E1M20copy.R is the rundeck file that is created in the "decks" directory. It contains a namelist called INPUTZ:

```
&INPUTZ
   YEARI=1949,MONTHI=12,DATEI=1,HOURI=0, IYEAR1=1949 ! or earlier
   YEARE=1949,MONTHE=12,DATEE=2,HOURE=0,     KDIAG=13*0,
   ISTART=2,IRANDI=0, YEARE=1949,MONTHE=12,DATEE=1,HOURE=1,
 /
```

As is this instructs the model to run for 1hour. To run, say for 5 years, remove the last line (starting with ISTART and change the YEARE line as follows:

```
&INPUTZ
   YEARI=1949,MONTHI=12,DATEI=1,HOURI=0, IYEAR1=1949 ! or earlier
   YEARE=1954,MONTHE=12,DATEE=1,HOURE=0,     KDIAG=13*0,
 /
```

The .R file has plenty of documentation but if it is not enough more can be found here.

6.4 Visualization

Given the widespread use of netCDF format in the Earth Sciences community (and the availability of packages to display the data therein) it is not surprising that modelE has the option to produce netCDF output. In the E1M20copy.R file there will be a line that reads as follows:

```
POUT                                  ! post-processing output
```

This controls the binary format for the diagnostic post-processing. To output in netCDF format change POUT to POUT_netcdf. Note that this is a compile-time option which means you have to compile the model with the desired option.

After running the model for 5 years several "acc" files will be generated and these are not netcdf files. Note: An acc file contains "accumulated" diagnostics from the previous month. These acc files must be post-processed to create netcdf output. To do so three more steps are necessary

Go to your output directory (under decks simply "cd E1M20copy") and:

1) Create a file named Ipd with the following contents:

```
&&PARAMETERS

 &&END_PARAMETERS

 &INPUTZ
 ISTART=-1,QDIAG=.true.,KDIAG=13*0,
 &END
```

2) run E1M20copyln (yes there is an "ln" at the end).

3) Execute the following (assuming you ran a 5-year run as explained earlier, the file NOV1954.accE1M20copy will be that last file output by the model):

```
E1M20copy.exe -i Ipd NOV1954.accE1M20copy < Ipd
```

Note #1: this last command will post-process only the last monthly output and will generate several netCDF files (.nc extension) that can be viewed under grads or matlab.

Note #2: If the executable E1M20copy.exe was built to run in parallel the above command should read "mpirun -np 1 E1M20copy.exe ..."

6.4.1 Post-processed output

As mentioned earlier each month the program produces an 'acc' accumulated diagnostics file which contains all of the diagnostic information from

the previous month. The program 'pdE' (in the exec sub-directory) is an alternate entry point into the model that reads in any number of these files and a) creates a printout for the time period concerned and b) creates binary output of many more diagnostics. This can be used simply to recreate the monthly printout, but also to create longer term means (annual, seasonal, monthly climatologies etc.).

For example, to recreate the printout in cmrun/$RUNID;

```
for a single month: pdE $RUNID JAN1987.acc$RUNID
for all Januaries:  pdE $RUNID JAN*.acc$RUNID
for a whole year:   pdE $RUNID *1987.acc$RUNID
```

For pdE to work properly, the directory cmrun/$RUNID has to exist and contain at least $RUNIDln $RUNIDuln $RUNID.exe . The output files will end up in the PRESENT WORKING DIRECTORY which may be cmrun/$RUNID or any other directory; names and order of the inputfiles are irrelevant (as long as the format of the files is compatible with the model $RUNID).

It is possible to use pdE to average acc-files from several runs, e.g. average over an ensemble of runs. Although the numbers that are produced are fine, subroutine aPERIOD will not be able to create the proper labels: the runID will be taken from the last file that was read in and the number of runs averaged will be interpreted as successive years, so averaging years 1951-1955 of runs runA runB runC runD will produce the label ANN1951-1970.runD rather than ANN1951-1955.runA-D. Some titles will also suffer from that 'bug', but it should be easy to fix it manually afterwards.

Note that the output can be controlled (a little) by the settings in 'Ipd' (which is created if it does not yet exist in the present working directory). A number of files will be created whose names contain the accumulated time period. (monyear[-year] where mon is a 3-letter acronym for a period of 1-12 consecutive months).

```
monyear.PRT      the printout
monyear.j$RUNID  zonal budget pages (ASCII Aplot format)
monyear.jk$RUNID latitude-height binary file
monyear.il$RUNID longitude-height binary file
monyear.ij$RUNID lat-lon binary file
```

```
monyear.wp$RUNID Wave power binary file
monyear.oij$RUNID lat-lon binary file for ocean diagnostics
monyear.ojl$RUNID lat-depth binary file for ocean diagnostics
monyear.oht$RUNID lat ASCII file for ocean heat transports
```

which can be read using the appropriate graphical software.

Note: If you have problems running pdE go to the $RUNDIR directory and simply executed something like "[mpirun -np 1] E1M20copy.exe -i Ipd *1950.accE1M20one ¡ Ipd". This produces annual averaged files with names like ANN1950.ijkE1M20one.nc

6.5 Stand-alone Package

The CIB toolkit contains a stand-alone package that allows a quick installation of modelE linkung to the supporting libraries (ESMF and netCDF).

First install the package modelE.scratch.tgz:

```
cp /cib/inputdata/archives/modelE.scratch.tgz .
tar xfz modelE.scratch.tgz
```

then

```
cd modelE.scratch
```

You will see the following directories and files

```
baselibs
cmrun
decks
etc
exec
input
modelE
options.rc
out
README
setupmodelE
```

First open/edit options.rc to make changes to the modelE setup. The options.rc file contains a very small number of options to help drive the setup process.

```
NETCDFDIR:
ESMFDIR:
DECK: E1M20
COMPILER: intel
MPI: mvapich2
# NETCDFDIR = netCDF installation is/will be in this -full- path
# ESMFDIR = ESMF installation is/will be in this -full- path
# DECK = select deck [default is E1M20]. A copy, E1M20test, will be created.
# COMPILER = Options are [intel, gfortran]
# MPI = Options are : intel, openmpi, mvapich2
```

Note that in the above example DECK=E1M20.

Finally execute setupmodelE. When done follow the next three steps:

```
cd modelE/decks
cp /cib/models/archives/modele_env.bash .
cp /cib/models/archives/modele_run.j .
```

Now you will need to to edit modele_run.j and change guestX to your username. When done submit the job to the batch system. To submit the job we use the qsub command:

```
qsub modele_run.j
```

You can monitor the job via the qstat command:

```
qstat
```

Results of the setup run will be in

```
/cib/outputdata/guestX/modelE.scratch/out/E1M20copy
```

Look at E1M20test.PRT, look at bottom for the following:

```
 >> Terminated normally (reached maximum time) <<
```

6.5.1 Case Study

The setup described in the previous section is intended to produce an experimental setting using the E1M20 deck that, upon execution of gmake setup, will produce a subdirectory out/E1M20copy that has run for one hour, that is, it has been 'setup'. The E1M20 deck is the simplest run deck in modelE and it is a good idea to try it first. For a more interesting case study we use the E4M20 deck. E4M20 is a low-resolution version of the atmosphere-only AR5 model (but without interactive chemistry or aerosols). To use this deck simply edit the options.rc file, specify DECK=E4M20 and re-run. Additionally you may want to edit the E4M20copy.R run deck file that will be generated under decks/. Edit so that the model runs for at least 30 days and produces netCDF output. See sections 6.3 and 6.4 respectively, for details on how to make those changes.

The E4M20 case study can be used as a benchmarking suite to port the model among platforms and to examine model reproducibility (see for example this document) and to some extent performance (see this document).

Chapter 7

WRF

7.1 Introduction

The WRF ARW model is a fully compressible, nonhydrostatic model (with a hydrostatic option). Its vertical coordinate is a terrain-following hydrostatic pressure coordinate. The grid staggering is the Arakawa C-grid. The model uses higher order numerics. This includes the Runge-Kutta 2nd and 3rd order time integration schemes, and 2nd to 6th order advection schemes in both horizontal and vertical directions. It uses a time-split small step for acoustic and gravity-wave modes. The dynamics conserves scalar variables. The WRF ARW model code contains several initialization programs (ideal.exe and real.exe), a numerical integration program (wrf.exe), and a program to do one-way nesting (ndown.exe). The WRF ARW model supports a variety of capabilities. These include:

- Real-data and idealized simulations
- Various lateral boundary condition options for both real-data and idealized simulations
- Full physics options
- Non-hydrostatic and hydrostatic (runtime option)
- One-way, two-way nesting and moving nest
- Applications ranging from meters to thousands of kilometers

For more information on the model, please check the following links:

- http://www.mmm.ucar.edu/wrf/OnLineTutorial/index.htm
- http://www.mmm.ucar.edu/wrf/users/docs/user_guide_V3/contents.html

It is assumed that we will use here WRF v3 and plan to run the Real Data case.

7.2 Obtaining the Code

The source code for the WRF ARW model is available from the download page:

http://www.mmm.ucar.edu/wrf/users/download/get_source.html

You need to register (it is free) to be able to obtain the code. You will need the following files:

- WRFV3.TAR.gz
- WPS.TAR.gz

7.3 Configuring and building WRF model

7.3.1 Create a working directory

mkdir myWorkDir

- for csh: setenv workDir path_to_myWorkDir
- for bashrc: export workDir=path_to_myWorkDir

Now untar the files:

- cd workDir

- tar xvfz WRFV3.TAR.gz
- tar xvfz WPS.TAR.gz

This will create the directories: WRFV3/ and WPS/.

- for csh:

  setenv wrfDir $workDir/WRFV3
  setenv wpsDir $workDir/WPS

- for bash:

  export wrfDir=$workDir/WRFV3
  export wpsDir=$workDir/WPS

Also provide the path where to find the necessary input data:

- for csh: setenv dataDir path_to_inputData
- for bash: export dataDir=path_to_inputData

The geographical input data set is located in:

- /share/LIBRARIES/WRF_GEOG

Also make sure that your ulimit setting is set to unlimited!

- ulimit u unlimited

7.3.2 Initial Settings

In order to compile the WRF model, we must first setup the environment variables needed. Insert the following into your .bashrc file:

```
# NetCDF Library
export NETCDF=/cib/libraries/netcdf/3.6.2_intel11
export PATH=$NETCDF/bin:$PATH
export MANPATH=$NETCDF/share/man:$MANPATH


# Settings for building WRF
module load comp/intel-11.1
module load mpi/mvapich2-1.5/intel-11.1
module load mkl/10.1.2
export BASEDIR=/cib/libraries/baselibs/3.2.0_intel11_mvapich2
export PATH=${BASEDIR}/Linux/bin:${PATH}
export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:${BASEDIR}/${ARCH}/lib
export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:/cib/libraries/libpng/1.4.3/lib/


# NCL Library
export NCARG_ROOT=/cib/libraries/ncl/5.2.0
export PATH=${NCARG_ROOT}/bin:$PATH
export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:/cib/libraries/gfortran/4.1.2/lib64
```

7.4 Compiling WRF

- cd $wrfDir

Step 1:

- ./configure

Make the following selections:

- Linux x86_64 i486 i586 i686, ifort compiler with icc (dmpar)
- Compile for nesting? (1=basic, 2=preset moves, 3=vortex following): 1

Step 2:

Edit the file configure.wrf to delete the following expressions:

-f90=$(SFC) and -cc=$(SCC).

Step 3:

Compile a WRF ARW real data case by typing:

- ./compile em_real >& compile.log &

You can check the progress of the compilation by typing:

- tail -f compile.log

Check the compile.log file for any errors. If your compilation was successful, you should see these executables created in the main/ directory.

File	Description
Main/ndown.exe	Used for one-way nesting
Main/nup.exe	Upscaling - used with WRF-Var
main/real.exe	WRF initialization for real data cases
main/wrf.exe	WRF model integration

These executables will be linked from the main/ directory to the directories run/ and test/em_real/, which is where you will be running the code from.

7.5 Compiling WPS

cd $wpsDir

Step 1:

- ./configure

Make the following selections:

- PC Linux x86_64, Intel compiler DM parallel (Option #4)

Step 2:

Edit the file configure.wps to replace

```
-L/usr/X11R6/lib -lX11
```

with

```
-L/usr/X11R6/lib64 -lX11
```

also in order to use the Grib2 format you must replace

```
COMPRESSION_LIBS  =  -L/data3a/mp/gill/WPS_LIBS/local/lib \
                     -ljasper -lpng12 -lpng -lz

COMPRESSION_INC   =  -I/data3a/mp/gill/WPS_LIBS/local/include

NCARG_LIBS        =   -L$(NCARG_ROOT)/lib -lncarg \
                      -lncarg_gks -lncarg_c \
                      -L/usr/X11R6/lib64 -lX11
```

with

```
COMPRESSION_LIBS =   -L/cib/libraries/jasper/1.900.1/lib \
                     -L/cib/libraries/libpng/1.4.3/lib \
                     -ljasper -lpng12 -lpng -lz

COMPRESSION_INC  =    -I/cib/libraries/jasper/1.900.1/include \
                      -I/cib/libraries/libpng/1.4.3/include

NCARG_LIBS       =    -L$(NCARG_ROOT)/lib -lncarg -lncarg_gks \
                      -lncarg_c \
-L/cib/libraries/gfortran/4.1.2/lib/gcc/x86_64-redhat-linux/4.1.2 \
                       -L/usr/lib64 -lX11 -lgfortran
```

Step 3:

Compile WPS by typing:

- ./compile >& compile.log &

If the compilation is successfull, you should see the executables:

File	Description
geogrid.exe -> geogrid/src/geogrid.exe	Generate static data
metgrid.exe -> metgrid/src/metgrid.exe	Generate input data for WRFV2
ungrib.exe -> ungrib/src/ungrid.exe	Unpack GRIB data

The following utilities are also created in the directory util/:

File	Description
avg_tsfc.exe	Computes daily mean surface temperature from intermediate files. Recommended for using with the 5-layer soil model (sf_surface_physics = 1) in WRF
calc_ecmwf_p.exe	
G1print.exe	List the contents of a GRIB1 file
G2print.exe	List the contents of a GRIB2 file
mod_levs.exe	Remove superfluous levels from 3-d fields in intermediate files
plotfmt.exe	Plot intermediate files (dependent on NCAR Graphics - if you don't have these libraries, plotfmt.exe will not compile correctly)
plotgrids.exe	Generate domain graphics. An exellent tool to configure domains before running geogrid.exe (dependent on NCAR Graphics - if you don't have these libraries, plotgrids.exe will not compile correctly)
rd_intermediate.exe	Read intermediate files

7.6 Steps to Run Geogrid

First, edit the file namelist.wps to set the variable geog_data_path, directory where the geographical data directories may be found:

- geog_data_path = $dataDir/geog

Next, setup the namelist.wps file to the proper resolution, location, date, etc. Run geogrid.exe by typing:

- ./geogrid.exe

If everything goes well, you will obtain the netCDF files with names such as geo_em_dxx.nc.

7.7 Steps to Run Ungrib

Step 1:

Link the proper Vtable:

- ln -sf ungrib/Variable_Tables/Vtable.GFS Vtable

Step 2:

Link the input GRIB or GRIB2 data:

- ./link_grib.csh path_to_GRIB_data

When done correctly you should have a list of links similar to the following:

- GRIBFILE.AAA
- GRIBFILE.AAB

- GRIBFILE.AAC
- GRIBFILE.AAD
- GRIBFILE.AAE
- GRIBFILE.AAF

Step 3:

Run UNGRIB

- ./ungrib.exe >& ungrib_data.log &

If ungrib runs successfully you should have files such as:

- FILE:2010-02-05_00
- FILE:2010-02-05_03
- FILE:2010-02-05_06
- FILE:2010-02-05_09
- FILE:2010-02-05_12

7.8 Steps to Run Metgrid

Run METGRID:

- ./metgrid.exe

The outputs from the run will be:

- met_em.d01.YYYY-MM-DD_DD:00:00.nc - one file for per time, and
- met_em.dxx.YYYY-MM-DD_DD:00:00.nc - one file for per nest, for the initial time only (met_em files for other times can be created, but are only needed for special FDDA runs).

7.9 Steps to Run WRF

Step 1:

Move to the WRF run/ or test/em_real directory:

- cd $wrfDir/run

Step 2:

Link the met_em files to this directory:

- ln -s $wpsDir/met_em*

Step 3:

Edit the input namelist file namelist.input.

Step 4:

Write a batch script to run real.exe in order to create the initial and boundary condition files: wrfinput_d01 and wrfbdy_d01. Here is a sample one (named run_real.job):

```
#!/bin/bash
#PBS -S /bin/bash
#PBS -N wrf_real
#PBS -l nodes=2:ppn=8
#PBS -l walltime=00:30:00
#PBS -j eo
cd $wrfDir/run       mpirun -hostfile $PBS_NODEFILE -np 16 ./real.exe
exit 0
```

Submit the job by typing:

- qsub run_real.job

Step 5:

Write a batch script to run wrf.exe. Here is a sample one (named run_wrf.job):

```
#!/bin/bash
#PBS -S /bin/bash
#PBS -N wrf_wrf
#PBS -l nodes=7:ppn=8
#PBS -l walltime=12:00:00
#PBS -j eo
cd $wrfDir/run
mpirun -hostfile $PBS_NODEFILE -np 56 ./wrf.exe      exit 0
```

Submit the job by typing:

- qsub run_wrf.job

The output file will be wrfout_dxx_initialDate that contains all the history records.

7.10 Case Study Description

The purpose of the case study is to test the limits of each machine and also provide data that can be used to prove identical results. The Feb 5th, 2010 Blizzard that produced copious amounts of snow for the DC region was picked. The following setup was used:

* Centered on Dulles International Airport (IAD)

* 48 hour simulation

* Triple Nested Domain

* Horizontal Resolution: 16, 4, 1 km

* Horizontal Grid Size: (198x191), (301x285), (357x345)

* 28 Vertical Levels
* Run with 48 nodes for each simulation

7.11 Run Case Study

Change into your user directory

```
cd /cib/outputdata/guestX
```

Copy over the WRF run directory

```
cp R /cib/models/wrf/wrf3.1/Case_Study_demo/run .
```

Move into the run directory

```
cd run
```

Change the path in run_wrf.job to reflect your current path (line 10)
Submit WRF script

```
qsub run_wrf.job
```

This will produce wrfout files for the 48 hour simulation.

Chapter 8

GFDL MOM4

The Modular Ocean Model (MOM) is a numerical representation of the ocean's hydrostatic primitive equations. It is designed primarily as a tool for studying the global ocean climate system, as well as capabilities for regional and coastal applications. MOM4 is the latest version of the GFDL ocean model whose origins date back to the pioneering work of Kirk Bryan and Mike Cox in the 1960s-1980s. It is developed and supported by researchers at NOAA's Geophysical Fluid Dynamics Laboratory (GFDL), with critical contributions also provided by researchers worldwide. In this chapter we provide particular information for how to download and run the code. Additional information will be found in the GFDL web site.

Note that the MOM4 source code and associated datasets are maintained at GForge. MOM4 users are required to register at the GFDL GForge location. Users need to register only once to get both the source code and datasets of MOM4. Registered users then need to request access to the relevant project (MOM4p1 is the most recent MOM4 project).

For an overview of the main characteristics of MOM4 please refer to [2].

8.1 Configuring and building

After you do CVS checkout successfully a directory will be created (refer to the manual [mom4_manual.html under doc/] for details). For convenience, this directory will be referred to as the ROOT directory. A README file in the ROOT directory will tell you the contents of each subdirectory under

ROOT.

MOM4 requires that NetCDF and MPI libraries be installed on users' platform. MOM4 tests are provided in the exp directory and are divided in two types, both using the GFDL shared infrastructure (FMS) :

- Solo models : Run stand alone MOM4 Ocean model.
- Coupled models: Run MOM4 coupled with GFDL ice model (besides null versions of atmosphere and land models).

8.1.1 Solo Models

1: cd to exp and run mom4p1_solo_compile.csh first.

2: Modify the 'name' variable in the script mom4p1_solo_run.csh to be the name of the test you want to run. A list of available tests is included in the script.

3: Get the required input data for the test from GFDL ftp site. You can get the info by running the script mom4p1_solo_run.csh and following the instructions.

4: Run mom4p1_solo_run.csh

5: The results go into subdirectory name/workdir

Users may also want to change the following before starting compilation and execution:

```
set npes = number of processors used in the run
set days = the length of the run in days
set months = the length of the run in months
```

Those are the most basic settings for any run. Experienced users may go to the namelist section in the script to set the values for namelist variables. Details on namelists can be found in the corresponding Fortran module.

8.1.2 Coupled Models

Do the same steps above to mom4p1_coupled_compile.csh and mom4p1_coupled_run.csh

8.1.3 Input data/Boundary conditions

The input data needed to run the selected experiments (tests) that are included in this release are available via anonymous ftp at

`ftp://ftp.gfdl.noaa.gov/perm/MOM4/mom4p1_pubrel/exp.`

Note that data in ASCII, HISTORY, RESTART directories are NOT needed for running experiments. They are the outputs of the experiments and are provided for the purpose of comparing your results with results produced at GFDL. Tools are provided so that users can create data from scratch for their own experiments. For more details refer to ROOT/src/preprocessing.

8.2 Examining the output

To keep the runscript simple all output files of a model run will be in the work directory. There are three types of output files:

1: ascii file with .fms.out extension: the description of the setup of the run and verbose comments printed out during the run.

2: restart files in RESTART directory: the model fields necessary to initialize future runs of the model.

3: history files with .nc.tar extension: output of the model, both averaged over specified time intervals and snapshots.

The ascii file contains everything written to the screen during model execution. The total time for model execution as well as the times of separate modules are reported here. All .tar files should be decompressed for viewing. The decompress command is:

```
tar xvf filename.tar
```

Users will see result files in NetCDF format. Postprocessing tools such as Ferret, ncview, grads or matlab can be used to view data in these files. The outputs of the selected experiments are provided on GFDL ftp site [ftp://ftp.gfdl.noaa.gov/perm/MOM4/ mom4p1_pubrel/exp] for the purpose of comparing your results with results produced at GFDL.

8.3 MOM4 Test Cases

MOM4 is distributed with a suite of test cases. Below is an outline of the main test cases available with MOM4p1.

mom4_atlantic: This regional model tests the open boundary condition option in MOM4p1.

mom4_bowl: This experiment has a sector model with a bowl topography.

mom4_box: This experiment has a sector model with a flat bottom topography. This is the simplest test case.

mom4_box_channel: This idealized experiment has a channel at the southern portion of the domain.

mom4_core: This experiment is global with a tripolar grid with roughly "1-degree" resolution and 50 vertical levels. The ocean is coupled to the GFDL sea ice model. The configuration is forced with the Normal Year forcing from the Coordinated Ocean Reference Experiment (data developed by Bill Large and Stephen Yeager at NCAR). This is a large model, and it is similar (though not the same) to the ocean and ice configuration used for the GFDL IPCC simulations.

mom4_dome: This is an idealized overflow configuration useful for testing overflow schemes.

mom4_ebm: This is a global model configuration coupled to the GFDL energy balance atmosphere plus the GFDL ice model.

mom4_iom: This experiment is a regional Indian Ocean model setup during a modeling school in Bangalore, India during October 2004.

mom4_mk3p5: This is a global spherical coordinate model which is based on the configuration used by CSIRO Marine and Atmospheric Research in Aspendale, AUS.

mom4_symmetric_box: This is an idealized configuration that is symmetric about the equator and uses symmetric forcing. The simulation should thus remain symmetric about the equator.

mom4_torus: This is an idealized simulation that is periodic in the x and y directions. It is useful for testing tracer advection schemes.

CM2p1: This test case represents a release of the CM2.1 coupled climate model used by GFDL scientists for the IPCC AR4 assessment of climate. This test includes the ocean configuration, land model, atmospheric model, and sea ice model setup as in CM2.1. Note that the original CM2.1 used the MOM4.0 code, whereas the CM2p1 test instead uses MOM4p1. However, GFDL scientists have verified that the climate simulations are compatible.

On the CIB platform an additional script, *MOM4_setup.bash* is included to simplify the MOM4p1 installation using one of the above test cases. To use, copy the *MOM4_setup.*bash script to the *exp* directory. The *MOM4_setup.bash* is driven by a text file called *options.rc* with the following options:

```
NETCDFDIR: /opt/libraries/Baselibs/3.2.0_intel11_mvapich2/Linux
EXPTYPE: mom4p1_coupled
EXPNAME: atlantic1
COMPILER: ia64
MPI: mvapich2
# NETCDFDIR = Location of netCDF installation (full path)
# EXPTYPE = select experiment type [ default mom4p1_solo;
```

```
#                                      options mom4p1_ebm,
#                                              mom4p1_coupled,
#                                              CM2.1p1_dynamic]
# EXPNAME = for type mom4p1_solo [default box1, options box1,
#                                                 torus1, gyre1,
#                                                 mk3p51]
#                    mom4p1_ebm  [default mom4p1_ebm1]
#                    mom4p1_coupled  [default atlantic1,
#                                     options MOM_SIS_TOPAZ]
#                    CM2.1p1_dynamic [default CM2.1p1]
# COMPILER = Options are [ia64 (i.e. intel), gfortran]
# MPI = Options are : openmpi, mvapich2
```

MOM4_setup.bash combines all the steps (and scripts) described in section 8.1 into one and the user can, if desired, use the MOM4 scripts instead.

Chapter 9

FAQ and tips

9.1 Compiler settings

9.2 Links to documentation

Bibliography

[1] N. Collins, G. Theurich, C. DeLuca, M. Suarez, A. Trayanov, V. Balaji, P. Li, W. Yang, C. Hill, and A. da Silva. Design and implementation of components in the earth system modeling framework. *Int. J. High Perf. Comput. Appl.*, 19:341–350, 2005.

[2] S.M. Griffies et al. A technical guide to mom4: Gfdl ocean group technical report no. 5. *NOAA/Geophysical Fluid Dynamics Laboratory*, 2004.

[3] M.M. Rienecker et al. The geos-5 data assimilation system - documentation of versions 5.0.1, 5.1.0, and 5.2.0. *Technical Report Series on Global Modeling and Data Assimilation, NASA GSFC*, (27), 2008.

[4] G.A. Schmidt et al. Present day atmospheric simulations using giss modele: Comparison to in-situ, satellite and reanalysis data. *J. Climate*, 19:153–192, 2006.